Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №9**

**«Работа с типизированными файлами»**

**ПО «МДК 05.02 РАЗРАБОТКА КОДА ИНФОРМАЦИОННЫХ СИСТЕМ»**

Выполнил: студент учебной группы

ИСПк-204-52-00

Алембаев Артём Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

Цель работы: получить представление об организации хранения и использования информации посредством типизированных файлов, закрепить навыки создания пользовательского интерфейса.

Задание:

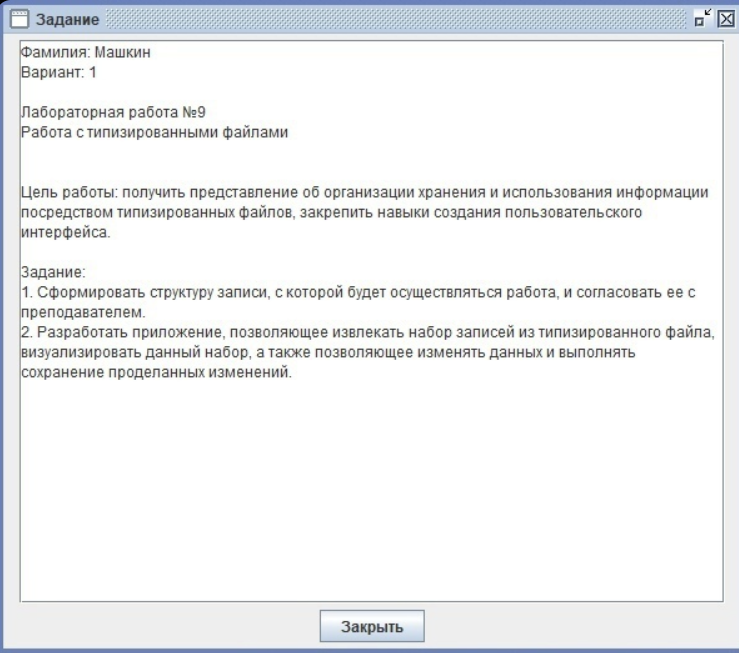


Рисунок 1 – Вариант задания

Описание алгоритма:

Тип Moto: Это запись, которая содержит информацию о мотоцикле, включая название, год выпуска, тип топлива (бензиновый или нет), мощность и цвет.

Глобальные переменные: mForm является экземпляром TmForm, а adres хранит путь к файлу данных.

Процедура FormCreate: Вызывается при создании формы и инициализирует таблицу mRec. Если файл данных moto.dat существует, процедура загружает данные из файла в таблицу.

Процедура mAddClick: Обрабатывает нажатие кнопки "Добавить". Открывает форму для ввода новой записи и, если пользователь сохраняет данные, добавляет новую строку в таблицу.

Процедура mDelClick: Обрабатывает нажатие кнопки "Удалить". Удаляет выбранную строку из таблицы после подтверждения пользователем.

Процедура mEditClick: Обрабатывает нажатие кнопки "Редактировать". Загружает данные выбранной строки в форму редактирования и, если пользователь сохраняет изменения, обновляет данные в таблице.

Процедура FormClose: Вызывается при закрытии формы. Сохраняет все данные из таблицы обратно в файл moto.dat.

Код программы:

unit mainmod;

{$mode objfpc}{$H+}

interface

uses

Classes, SysUtils, Forms, Controls, Graphics, Dialogs, ExtCtrls, Buttons,

Grids, createRec;

type

{ TmForm }

TmForm = class(TForm)

mAdd: TBitBtn;

mEdit: TBitBtn;

mDel: TBitBtn;

Panel: TPanel;

mRec: TStringGrid;

procedure FormClose(Sender: TObject; var CloseAction: TCloseAction);

procedure FormCreate(Sender: TObject);

procedure mAddClick(Sender: TObject);

procedure mDelClick(Sender: TObject);

procedure mEditClick(Sender: TObject);

procedure PanelClick(Sender: TObject);

private

public

end;

type

Moto = record

Name: string[100];

Year: integer;

Petrol: boolean;

Power: integer;

Color: string[30];

end;

var

mForm: TmForm;

adres: string;

implementation

{$R \*.lfm}

{ TmForm }

procedure TmForm.FormCreate(Sender: TObject);

var

newMoto: Moto; //для очередной записи

f: file of Moto; //файл данных

i: integer; //счетчик цикла

begin

adres:= ExtractFilePath(ParamStr(0));

mRec.Cells[0, 0]:= 'Название';

mRec.Cells[1, 0]:= 'Год';

mRec.Cells[2, 0]:= 'Бензиновый';

mRec.Cells[3, 0]:= 'Мощность';

mRec.Cells[4, 0]:= 'Цвет';

mRec.ColWidths[0]:= 200;

mRec.ColWidths[1]:= 100;

mRec.ColWidths[2]:= 100;

mRec.ColWidths[3]:= 100;

mRec.ColWidths[4]:= 129;

//если файла данных нет, просто выходим:

if not FileExists(adres + 'moto.dat') then exit;

//иначе файл есть, открываем его для чтения и

//считываем данные в сетку:

try

AssignFile(f, adres + 'moto.dat');

Reset(f);

//теперь цикл - от первой до последней записи сетки:

while not Eof(f) do begin

//считываем новую запись:

Read(f, newMoto);

//добавляем в сетку новую строку, и заполняем её:

mRec.RowCount:= mRec.RowCount + 1;

mRec.Cells[0, mRec.RowCount-1]:= newMoto.Name;

mRec.Cells[1, mRec.RowCount-1]:= IntToStr(newMoto.Year);

if (newMoto.Petrol) then mRec.Cells[2, mRec.RowCount-1]:= 'Да'

else mRec.Cells[2, mRec.RowCount-1]:='Нет';

mRec.Cells[3, mRec.RowCount-1]:= FloatToStr(newMoto.Power);

mRec.Cells[4, mRec.RowCount-1]:= newMoto.Color;

end;

finally

CloseFile(f);

end;

end;

procedure TmForm.mAddClick(Sender: TObject);

begin

//очищаем поля, если там что-то есть:

createRecord.eName.Text:= '';

createRecord.eYear.Text:= '';

createRecord.ePower.Text:= '';

//устанавливаем ModalResult редактора в mrNone:

createRecord.ModalResult:= mrNone;

//теперь выводим форму:

createRecord.ShowModal;

//если пользователь ничего не ввел - выходим:

if (createRecord.eName.Text= '') or (createRecord.eYear.Text= '') or (createRecord.ePower.Text= '') then exit;

//если пользователь не нажал "Сохранить" - выходим:

if createRecord.ModalResult <> mrOk then exit;

//иначе добавляем в сетку строку, и заполняем её:

mRec.RowCount:= mRec.RowCount + 1;

mRec.Cells[0, mRec.RowCount-1]:= createRecord.eName.Text;

mRec.Cells[1, mRec.RowCount-1]:= createRecord.eYear.Text;

if (createRecord.ePetrol.Checked) then mRec.Cells[2, mRec.RowCount-1]:= 'Да'

else mRec.Cells[2, mRec.RowCount-1]:='Нет';

mRec.Cells[3, mRec.RowCount-1]:= createRecord.ePower.Text;

mRec.Cells[4, mRec.RowCount-1]:= createRecord.eColor.Text;

end;

procedure TmForm.mDelClick(Sender: TObject);

begin

//если данных нет - выходим:

if mRec.RowCount = 1 then exit;

//иначе выводим запрос на подтверждение:

if MessageDlg('Требуется подтверждение',

'Вы действительно хотите удалить запись "' +

mRec.Cells[0, mRec.Row] + '"?',

mtConfirmation, [mbYes, mbNo, mbIgnore], 0) = mrYes then

mRec.DeleteRow(mRec.Row);

end;

procedure TmForm.mEditClick(Sender: TObject);

begin

//если данных в сетке нет - просто выходим:

if mRec.RowCount = 1 then exit;

//иначе записываем данные в форму редактора:

createRecord.eName.Text:= mRec.Cells[0, mRec.Row];

createRecord.eYear.Text:= mRec.Cells[1, mRec.Row];

if (mRec.Cells[2, mRec.RowCount-1] = 'Да') then createRecord.ePetrol.Checked := True

else createRecord.ePetrol.Checked := False;

createRecord.ePower.Text:= mRec.Cells[3, mRec.Row];

createRecord.eColor.Text:= mRec.Cells[4, mRec.Row];

//устанавливаем ModalResult редактора в mrNone:

createRecord.ModalResult:= mrNone;

//теперь выводим форму:

createRecord.ShowModal;

//сохраняем в сетку возможные изменения,

//если пользователь нажал "Сохранить":

if createRecord.ModalResult = mrOk then begin

mRec.Cells[0, mRec.Row]:= createRecord.eName.Text;

mRec.Cells[1, mRec.Row]:= createRecord.eYear.Text;

if (createRecord.ePetrol.Checked) then mRec.Cells[2, mRec.RowCount-1]:= 'Да'

else mRec.Cells[2, mRec.RowCount-1]:='Нет';

mRec.Cells[3, mRec.Row]:= createRecord.ePower.Text;

mRec.Cells[4, mRec.Row]:= createRecord.eColor.Text;

end;

end;

procedure TmForm.PanelClick(Sender: TObject);

begin

end;

procedure TmForm.FormClose(Sender: TObject; var CloseAction: TCloseAction);

var

newMoto: Moto; //для очередной записи

f: file of Moto; //файл данных

i: integer; //счетчик цикла

begin

//если строки данных пусты, просто выходим:

if mRec.RowCount = 1 then exit;

//иначе открываем файл для записи:

try

AssignFile(f, adres + 'moto.dat');

Rewrite(f);

//теперь цикл - от первой до последней записи сетки:

for i:= 1 to mRec.RowCount-1 do begin

//получаем данные текущей записи:

newMoto.Name:= mRec.Cells[0, i];

newMoto.Year:= StrToInt(mRec.Cells[1, i]);

if (mRec.Cells[2, mRec.RowCount-1] = 'Да') then newMoto.Petrol := True

else newMoto.Petrol := False;

newMoto.Power:= StrToInt(mRec.Cells[3, i]);

newMoto.Color:= mRec.Cells[4, i];

//записываем их:

Write(f, newMoto);

end;

finally

CloseFile(f);

end;

end;

end.

Результат выполнения программ:

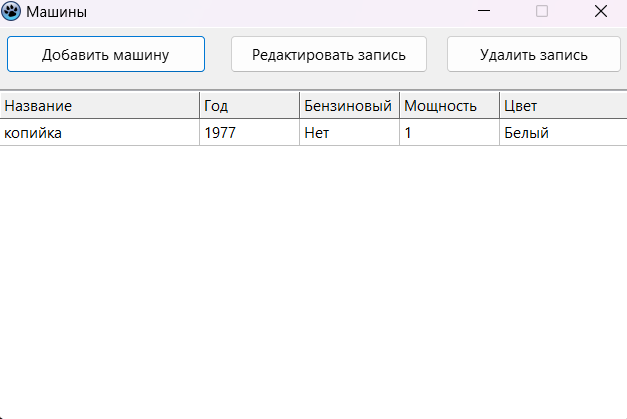


Рисунок 2 – Итоговый результат

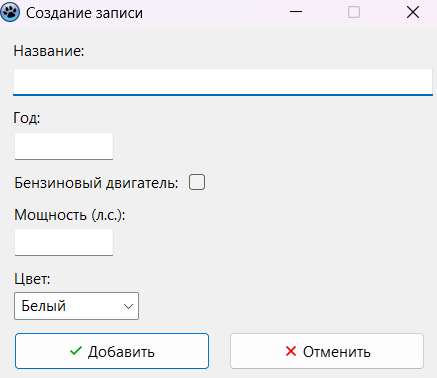


Рисунок 3 – Итоговый результат

Вывод: в ходе выполнения лабораторной работы были получено представление об организации хранения и использования информации посредством типизированных файлов, закреплены навыки создания пользовательского интерфейса.